

Partie 5 Timer



khalil.hachicha@gmail.com

Timer



- Un Timer est un compteur programmable qui est incrémenté à chaque cycle d'une horloge interne au microcontrôleur
- Quelle est son utilité?
 - Générer des délais (à la place d'une boucle for)
 - Mesurer des temps
 - Générer des signaux en fonction des délais ou des temps mesurés

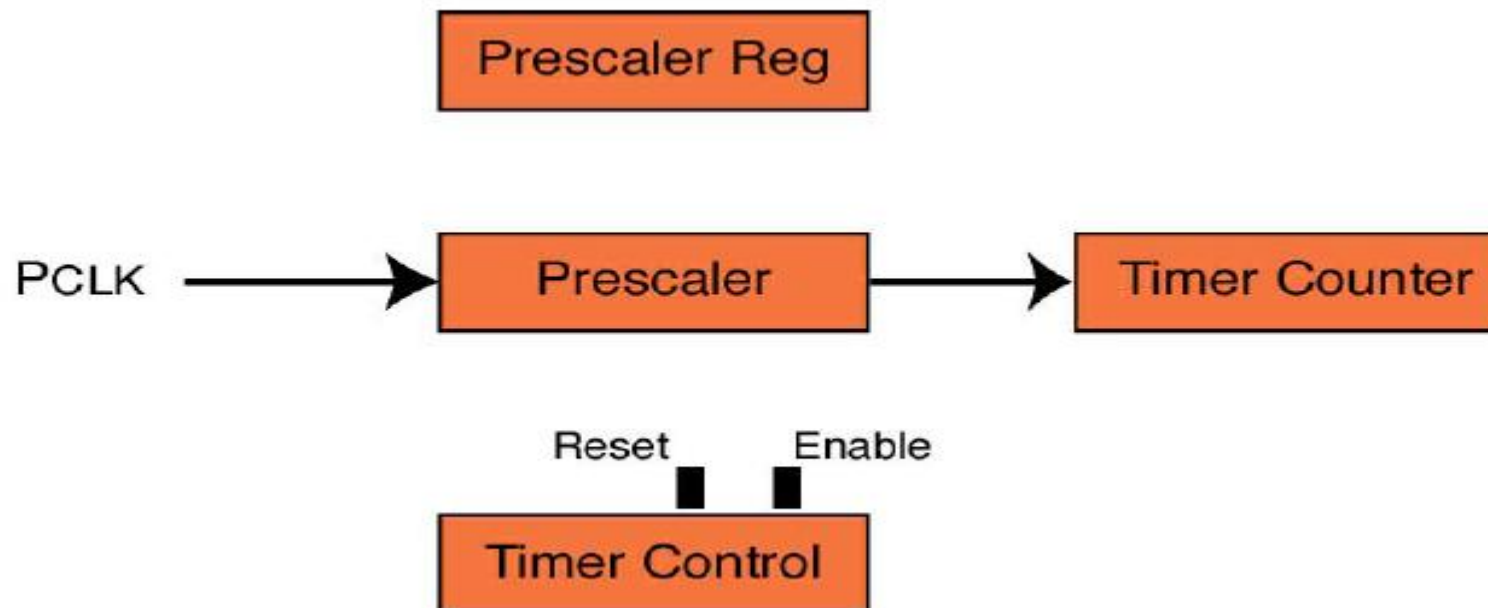
Timers et LPC2378



- Le LPC2378 possède plusieurs timers
 - 4 Timers 32 bits à usage général (T0, T1, T2, T3)
 - 4 entrées de Capture par timer
 - 4 registres de comparaison par timer
 - 1 Générateur de signaux PWM (avec Timer 32 bits)
 - 1 Watchdog Timer de 32 bits

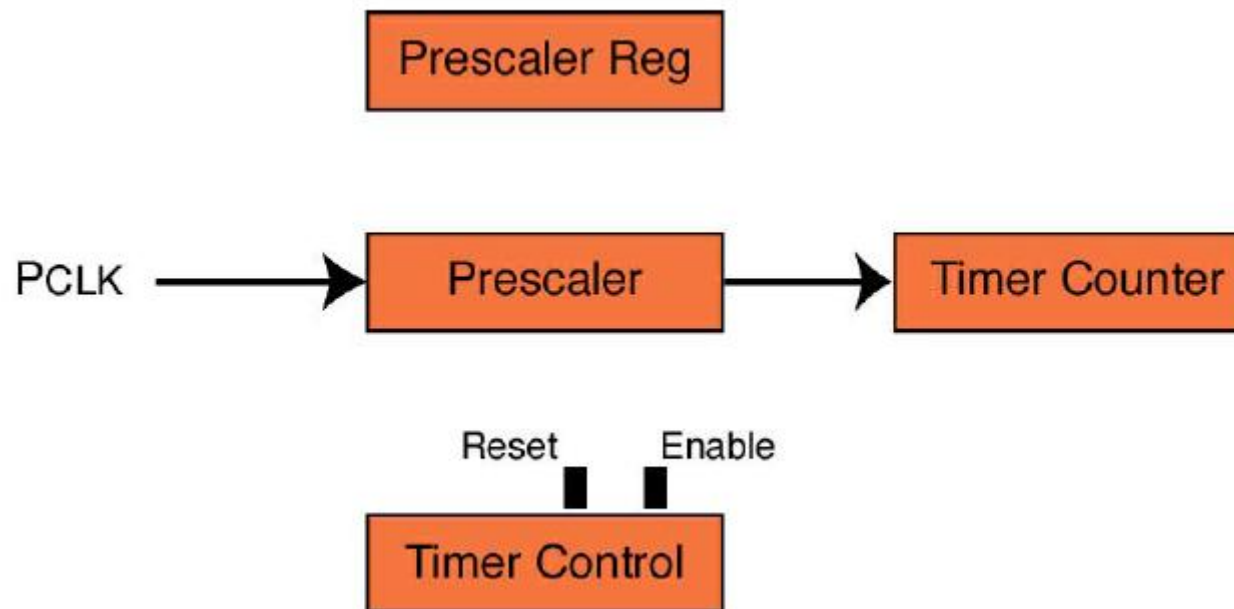
Architecture – Timers LPC2378

- PCLK: Horloge périphérique
 - Différente de celle du processeur (CCLK)
 - La valeur de PCLK est proportionnelle à celle de CCLK
 - Par défaut, $F(\text{PCLK}) = F(\text{CCLK}) / 4$
 - Modifiable à l'aide des registres PCLKSEL0 et PCLKSEL1



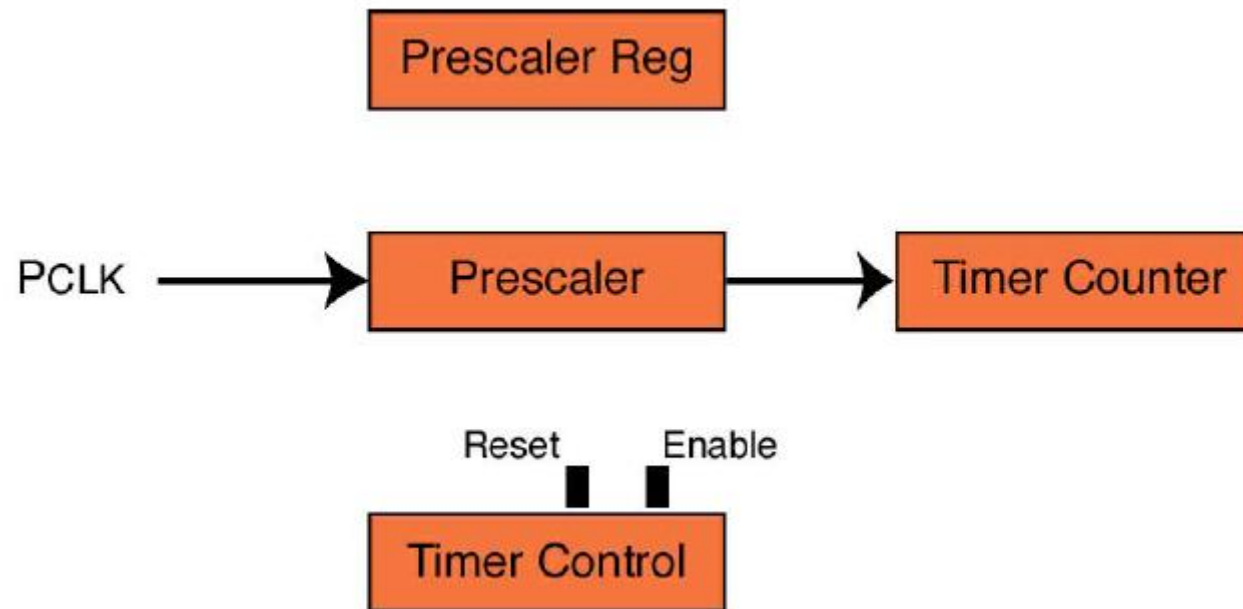
Architecture – Timers LPC2378

- Prescaler et Prescal Register
 - Précompteurs
 - Prescaler s'incrémente à chaque cycle de PCLK
 - Lorsque le Prescaler atteint la valeur du Prescaler Register, le Timer s'incrémente



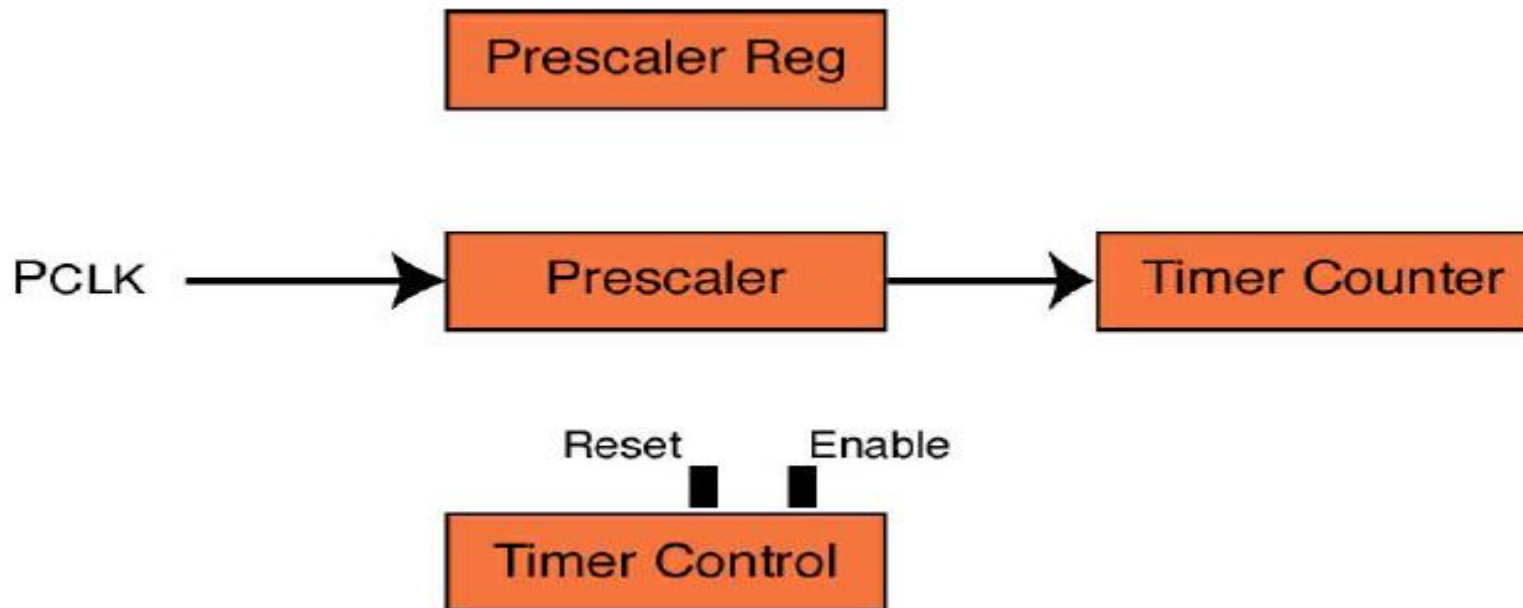
Architecture – Timers LPC2378

- Prescaler et Prescal Register
 - Au Reset, Prescaler = Prescaler Register = 0
 - Par défaut, le Timer s'incrémente à chaque cycle de PCLK



Architecture – Timers LPC2378

- Registres Correspondants
 - Timer Counter : TxTC (Lecture seule)
 - Prescaler : TxPC (Lecture seule)
 - Prescaler Register : TxPR
 - $x = 0, 1, 2$ ou 3



Registre de Commande

- Registre TxTCR (x=0,1,2 ou 3)
 - Bit 0: Démarrage/Arrêt du Compteur
 - Bit 1: Reset du Compteur

Table 430: Timer Control Register (TCR, TIMERN: TnTCR - addresses 0xE000 4004, 0xE000 8004, 0xE007 0004, 0xE007 4004) bit description

| Bit | Symbol | Description | Reset Value |
|-----|----------------|---|-------------|
| 0 | Counter Enable | When one, the Timer Counter and Prescale Counter are enabled for counting. When zero, the counters are disabled. | 0 |
| 1 | Counter Reset | When one, the Timer Counter and the Prescale Counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to zero. | 0 |

Match (Comparaison)



- Chaque Timer possède 4 Match Register
 - TxMR0, TxMR1, TxMR2, TxMR3
- A chaque incrémentation du Timer Tx, le contenu de TxTC est comparé aux Match Registers
 - Si les deux valeurs sont égales, une action peut alors être entreprise
- Intérêt: Permet de déterminer la durée d'une temporisation

Actions et Match Registers

- Les actions entreprises sont définies par le Match Control Register TxMCR
- Selon sa valeur il est possible de
 - Demander une interruption
 - D'arrêter le Timer
 - De remettre à 0 la valeur du Timer
- Ces 3 actions sont indépendantes et peuvent avoir lieu simultanément
- Tout ceci n'intervient que si $TxTC = TxMRy$

Registre TxMCR

- Registre 16 bits (12 bits utiles) - 3 bits par Timer
 - Pour TxMR0
 - Bit 0: Demande d'interruption
 - Bit 1: Reset
 - Bit 2: Stop

| Bit | Symbol | Value | Description | Reset Value |
|-----|--------|-------|---|-------------|
| 0 | MR0I | 1 | Interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC. | 0 |
| | | 0 | This interrupt is disabled | |
| 1 | MR0R | 1 | Reset on MR0: the TC will be reset if MR0 matches it. | 0 |
| | | 0 | Feature disabled. | |
| 2 | MR0S | 1 | Stop on MR0: the TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC. | 0 |
| | | 0 | Feature disabled. | |

Même fonctionnement pour TxMR1 (Bits 3-5), TxMR2 (Bits 6-8) et TxMR3 (Bits 9-11)

Durée des Temporisations



- Comment calculer le temps qu'il faudra au compteur du Timer pour arriver à la valeur d'un Match Register (T0MR0 par exemple)

$$\text{Temps} = \text{T0MR0} / \text{F(PCLK)}$$

- De même, comment déterminer la valeur à écrire dans T0MR0 pour effectuer une temporisation d'une durée T

$$\text{T0MR0} = T \times \text{F(PCLK)}$$

Registre d'Interruption du Timer

- Registre 8 bits TxIR
 - Chaque bit correspond à une source d'interruption
 - Au niveau du contrôleur, ces 8 sources sont reliées au même numéro de canal (4 pour Timer 0)

| Bit | Symbol | Description | Reset Value |
|-----|---------------|---|-------------|
| 0 | MR0 Interrupt | Interrupt flag for match channel 0. | 0 |
| 1 | MR1 Interrupt | Interrupt flag for match channel 1. | 0 |
| 2 | MR2 Interrupt | Interrupt flag for match channel 2. | 0 |
| 3 | MR3 Interrupt | Interrupt flag for match channel 3. | 0 |
| 4 | CR0 Interrupt | Interrupt flag for capture channel 0 event. | 0 |
| 5 | CR1 Interrupt | Interrupt flag for capture channel 1 event. | 0 |
| 6 | CR2 Interrupt | Interrupt flag for capture channel 2 event. | 0 |
| 7 | CR3 Interrupt | Interrupt flag for capture channel 3 event. | 0 |

- Si une source demande une interruption, son bit passe à 1
- Pour acquitter l'IRQ, il faut écrire un 1 sur le bit correspondant

Exemple

- Génération d'un signal périodique de période 2 ms et de rapport cyclique 50% sur la broche P2.0
 - Données: CCLK=48 MHz.

Procédure d'Initialisation

```
#include "LPC23xx.h"
void init_T0(void)
{
    FIO..DIR =
    VICVectAddr.. =
    VICIntEnable =
    T0MR.. =
    T0MCR =
    T0TCR =
}
```

Sous Programme d'Interruption



```
void          (void) __irq  
{
```

```
    T..IR =  
    VICVectAddr =  
}
```


Programme Principal



```
int main()
```

```
{
```

```
}
```